

**AMENDMENTS TO THE CLAIMS**

The listing of claims will replace all prior versions, and listings, of claims in the application:

**Listing of Claims:**

1. (Currently Amended) A computer implemented method for asymmetrically multithreading tasks in a computer environment that includes a plurality of dissimilar processors, said method comprising:  
  
scheduling operation of a plurality of first tasks adapted to be executed by a first type of processor from the plurality of dissimilar processors, the scheduling performed by a first scheduler that maintains a first run queue that includes data corresponding to the first tasks;  
  
in response to the scheduling operation of the plurality of first tasks by the first scheduler, inserting the plurality of first tasks in the first run queue;  
  
scheduling operation of a plurality of second tasks adapted to be executed by a second type of processor from the plurality of dissimilar processors, the scheduling performed by a second scheduler that is different than the first scheduler and maintains a second run queue that includes data corresponding to the second tasks, wherein the second scheduler schedules the second tasks concurrently with when the first scheduler schedules the first tasks;  
  
in response to the scheduling operation of the plurality of second tasks by the second scheduler, inserting the plurality of second tasks in the second run queue; and  
  
wherein the first scheduler and the second scheduler are both located on a first processor that is the first type of processor.
2. (Canceled)

3. (Previously Presented) The method as described in claim 1 wherein the scheduling operation of the plurality of first tasks is asymmetric to the scheduling operation of the plurality of second tasks.
4. (Previously Presented) The method as described in claim 1 further comprising:  
wherein a first identifier space corresponds to the first type of processor and  
wherein a second identifier space corresponds to the second type of processor;  
and  
wherein a first task list corresponds to the first type of processor and wherein a second task list corresponds to the second type of processor.
5. (Previously Presented) The method as described in claim 1 wherein the scheduling operation of the plurality second tasks further comprises:  
receiving a new task from the plurality of second tasks;  
identifying new task attributes corresponding to the new task;  
comparing the new task attributes with one or more scheduled task attributes, the scheduled task attributes corresponding to one or more scheduled tasks that are included in the second run queue; and  
performing the scheduling of the new task based upon the comparing.
6. (Original) The method as described in claim 5 wherein at least one of the new task attributes are selected from the group consisting of a policy and a priority.
7. (Previously Presented) The method as described in claim 1 further comprising:  
informing the second type of processor to load one of the second tasks in response to the scheduling.
8. (Original) The method as described in claim 1 wherein the computer environment includes a plurality of second type of processors, and wherein the

second scheduler maintains a plurality of second run queues, each of the plurality of second run queues corresponding to each of the plurality of second type of processors.

9. (Original) The method as described in claim 8 wherein the scheduling operation of the plurality of second tasks further comprises:

receiving a new task from the plurality of second tasks

analyzing a plurality of workloads that correspond to the plurality of second type of processors;

identifying an available second type of processor from the plurality of second type of processors in response to the analyzing; and

including the new task in the second run queue from the plurality of second run queues that corresponds to the available second type of processor.

10. (Previously Presented) The method as described in claim 1 wherein the first type of processor is a processing unit that operates in a trusted mode environment and wherein the second type of processor is a synergistic processing unit that operates in a non-trusted mode environment.

11. (Currently Amended) An information handling system comprising:

a plurality of dissimilar processors;

a memory accessible by the plurality of dissimilar processors;

one or more nonvolatile storage devices accessible by the plurality of dissimilar processors; and

an asymmetric multithreading tasking tool for scheduling tasks, the asymmetric multithreading tasking tool comprising software code effective to:

schedule operation of a plurality of first tasks adapted to be executed by a first type of processor from the plurality of dissimilar processors, the

scheduling performed by a first scheduler that maintains a first run queue located on the memory that includes data corresponding to the first tasks; in response to the scheduling operation of the plurality of first tasks by the first scheduler, insert the plurality of first tasks in the first run queue; schedule operation of a plurality of second tasks adapted to be executed by a second type of processor from the plurality of dissimilar processors, the scheduling performed by a second scheduler that is different than the first scheduler and maintains a second run queue that includes data corresponding to the second tasks, wherein the second scheduler schedules the second tasks concurrently with when the first scheduler schedules the first tasks;

in response to the scheduling operation of the plurality of second tasks by the second scheduler, insert the plurality of second tasks in the second run queue; and

wherein the first scheduler and the second scheduler are both located on a first processor that is the first type of processor.

12. (Canceled)
13. (Previously Presented) The information handling system as described in claim 11 wherein the scheduling operation of the plurality of first tasks is asymmetric to the scheduling operation of the plurality of second tasks.
14. (Previously Presented) The information handling system as described in claim 11 wherein the software code is further effective to:
 

wherein a first identifier space corresponds to the first type of processor and wherein a second identifier space corresponds to the second type of processor; and

wherein a first task list corresponds to the first type of processor and wherein a second task list corresponds to the second type of processor.

15. (Previously Presented) The information handling system as described in claim 11 wherein the software code is further effective to:  
  
receive a new task from the plurality of second tasks;  
  
identify new task attributes corresponding to the new task;  
  
compare the new task attributes with one or more scheduled task attributes, the scheduled task attributes corresponding to one or more scheduled tasks that are included in the second run queue; and  
  
perform the scheduling of the new task based upon the comparing.
16. (Original) The information handling system as described in claim 15 wherein at least one of the new task attributes are selected from the group consisting of a policy and a priority.
17. (Previously Presented) The information handling system as described in claim 11 wherein the software code is further effective to:  
  
inform the second type of processor to load one of the second tasks in response to the scheduling.
18. (Previously Presented) The information handling system as described in claim 11 wherein the computer environment includes a plurality of second type of processors, and wherein the second scheduler maintains a plurality of second run queues, each of the plurality of second run queues corresponding to each of the plurality of second type of processors.
19. (Original) The information handling system as described in claim 18 wherein the software code is further effective to:

receive a new task from the plurality of second tasks

analyze a plurality of workloads that correspond to the plurality of second type of processors;

identify an available second type of processor from the plurality of second type of processors in response to the analyzing; and

include the new task in the second run queue from the plurality of second run queues that corresponds to the available second type of processor.

20. (Previously Presented) The information handling system as described in claim 11 wherein the first type of processor is a processing unit that operates in a trusted mode environment and wherein the second type of processor is a synergistic processing unit that operates in a non-trusted mode environment.
21. (Currently Amended) A computer program product stored on a computer operable media for asymmetrically multithreading tasks in a computer environment that includes a plurality of dissimilar processors, said computer program product comprising:
 

means for scheduling operation of a plurality of first tasks adapted to be executed by a first type of processor from the plurality of dissimilar processors, the scheduling performed by a first scheduler that maintains a first run queue that includes data corresponding to the first tasks;

in response to the scheduling operation of the plurality of first tasks by the first scheduler, means for inserting the plurality of first tasks in the first run queue;

means for scheduling operation of a plurality of second tasks adapted to be executed by a second type of processor from the plurality of dissimilar processors, the scheduling performed by a second scheduler that is different than the first scheduler and maintains a second run queue that includes data

corresponding to the second tasks, wherein the second scheduler schedules the second tasks concurrently with when the first scheduler schedules the first tasks;

in response to the scheduling operation of the plurality of second tasks by the second scheduler, means for inserting the plurality of second tasks in the second run queue; and

wherein the first scheduler and the second scheduler are both located on a first processor that is the first type of processor.

22. (Canceled)
23. (Previously Presented) The computer program product as described in claim 21 wherein the scheduling operation of the plurality of first tasks is asymmetric to the scheduling operation of the plurality of second tasks.
24. (Previously Presented) The computer program product as described in claim 21 further comprising:

wherein a first identifier space corresponds to the first type of processor and wherein a second identifier space corresponds to the second type of processor; and

wherein a first task list corresponds to the first type of processor and wherein a second task list corresponds to the second type of processor.
25. (Previously Presented) The computer program product as described in claim 21 wherein the scheduling operation of the plurality second tasks further comprises:

means for receiving a new task from the plurality of second tasks;

means for identifying new task attributes corresponding to the new task;

means for comparing the new task attributes with one or more scheduled task attributes, the scheduled task attributes corresponding to one or more scheduled tasks that are included in the second run queue; and

means for performing the scheduling of the new task based upon the comparing.

26. (Original) The computer program product as described in claim 25 wherein at least one of the new task attributes are selected from the group consisting of a policy and a priority.
27. (Previously Presented) The computer program product as described in claim 21 further comprising:

means for informing the second type of processor to load one of the second tasks in response to the scheduling.
28. (Original) The computer program product as described in claim 21 wherein the computer environment includes a plurality of second type of processors, and wherein the second scheduler maintains a plurality of second run queues, each of the plurality of second run queues corresponding to each of the plurality of second type of processors.
29. (Original) The computer program product as described in claim 28 wherein the scheduling operation of the plurality of second tasks further comprises:

means for receiving a new task from the plurality of second tasks

means for analyzing a plurality of workloads that correspond to the plurality of second type of processors;

means for identifying an available second type of processor from the plurality of second type of processors in response to the analyzing; and

means for including the new task in the second run queue from the plurality of second run queues that corresponds to the available second type of processor.



30. (Previously Presented) The computer program product as described in claim 21 wherein the first type of processor is a processing unit that operates in a trusted mode environment and wherein the second type of processor is a synergistic processing unit that operates in a non-trusted mode environment.